Semantic Segmentation for Full-Waveform LiDAR Data Using Local and Hierarchical Global Feature Extraction

Takayuki Shinohara
Tokyo Institute of Technology
Yokohama, Kanagawa, Japan
shinohara.t.af@m.titech.ac.jp

Haoyi Xiu
Tokyo Institute of Technology
Yokohama, Kanagawa, Japan
xiu.h.aa@m.titech.ac.jp

Masashi Matsuoka
Tokyo Institute of Technology
Yokohama, Kanagawa, Japan
matsuoka.m.ab@m.titech.ac.jp

ABSTRACT

During the last few years, in the field of computer vision, sophisticated deep learning methods have been developed to accomplish semantic segmentation tasks of 3D point cloud data. Additionally, many researchers have extended the applicability of these methods, such as PointNet or PointNet++, beyond semantic segmentation tasks of indoor scene data to large-scale outdoor scene data observed using airborne laser scanning systems equipped with light detection and ranging (LiDAR) technology. Most extant studies have only investigated geometric information (x, y, and z or longitude, latitude, and height) and have omitted rich radiometric information. Therefore, we aim to extend the applicability of deep learning-based model from the geometric data into radiometric data acquired with airborne full-waveform LiDAR without converting the waveform into 2D images or 3D voxels. We simultaneously train two models: a local module for local feature extraction and a global module for acquiring wide receptive fields for the waveform. Furthermore, our proposed model is based on waveform-aware convolutional techniques. We evaluate the effectiveness of the proposed method using benchmark large-scale outdoor scene data. By integrating the two outputs from the local module and the global module, our proposed model had achieved higher mean recall value 0.92 than previous methods and higher F1 scores for all six classes than the other 3D Deep Learning method. Therefore, our proposed network consisting of the local and global module successfully resolves the semantic segmentation task of full-waveform LiDAR data without requiring expert knowledge.

CCS CONCEPTS

• **Computing methodologies** → **Scene understanding**.

KEYWORDS

Full-waveform LiDAR, Neural networks, Semantic segmentation, Supervised learning

ACM Reference Format:
Takayuki Shinohara, Haoyi Xiu, and Masashi Matsuoka. 2020. Semantic Segmentation for Full-Waveform LiDAR Data Using Local and Hierarchical Global Feature Extraction. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3397536.3422209

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8019-5/20/11…$15.00
https://doi.org/10.1145/3397536.3422209

1 INTRODUCTION

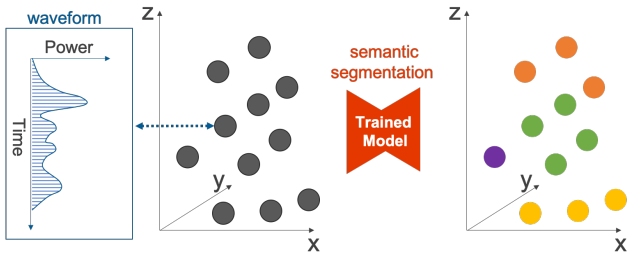


Figure 1: Illustration of our method. Each point includes geometric information (i.e., x, y, z), and waveform consists of the travel time and power of return signals. The trained model predicts the class of each point

Semantic segmentation for 3D data is a challenging task because 3D point clouds have spatially irregular structures (called *non-Euclidian* data). With the development of deep learning techniques in the field of computer vision, many researchers have investigated deep learning-based methods for 3D point clouds [19, 29, 39, 46, 49]. The most conventional method is a 2D convolutional neural network (CNN) for image recognition to classify 2D images projected from point clouds [46, 53]. These methods usually require the calculation of additional handcrafted features of point clouds when they project 2D images from 3D point clouds. However, these methods are not performed because of information loss during 3D projection to a 2D image. In more recent studies, attempts have been made to handle 3D information represented as voxel data [13]. Voxel-based methods use 3D convolution for regular 3D grid data (called voxels) converted from the point cloud. In this case, when point clouds are converted to voxels, classification performance is adversely affected because of information loss. However, some studies have applied CNN techniques on irregular point clouds [39, 41, 44, 49]. These methods have shown state-of-the-art performance on several point cloud semantic segmentation benchmarks.

With the rapid development of 3D deep learning methods, 3D point cloud data observed in large outdoor scenes using airborne laser scanning (ALS) devices have become increasingly accessible to computer-vision applications. Light detection and ranging (LiDAR)

provides reliable 3D geometric information as 3D point clouds, including x, y, and z or $latitude$, $longitude$, and $height$ information, calculated from the sensor position and return times of emitted pulses. Additionally, ALS plays an important role in many applications, such as topographic mapping, forest monitoring [5, 27], power-line detection [4, 9, 34, 52], road detection and planning, and 3D-building detection [16, 45]. Recently, full-waveform LiDAR has been widely used for ALS observations because it can discretely record return values, and the recorded shapes can represent surface characteristics. Most 3D deep learning methods have focused on 3D geometric information while omitting rich radiometric waveforms. However, in some of the previous studies, deep learning-based analysis for full-waveform LiDAR data, including point clouds as geometric information and waveforms as radiometric information, has been investigated. Zorti et al. [55] proposed a fully convolutional network (FCN)-based semantic segmentation method for 2D images converted from waveforms. The method of Zorti et al. [55] suffered from occlusions and information loss caused by the translation of 3D data to 2D image data. However, a method to directly handle the full-waveform LiDAR data was presented by Shinohara et al. [33]. Shinohara et al. [33] showed the effectiveness of a spatial learning method for full-waveform LiDAR data with point clouds and waveforms using representation learning with an autoencoder-based network called the full-waveform net autoencoder (FWNetAE). Specifically, instead of inputting each waveform individually into the deep learning model, FWNetAE uses a range of waveforms and coordinates information simultaneously to input a certain range of full-waveform LiDAR data into a deep learning model based on PointNet. FWNetAE based on PointNet enables spatial feature extraction and has a better feature extraction performance than learning individual waveforms. However, FWNetAE only showed the power of feature extraction. Thus, we must examine a more concrete method for semantic segmentation tasks for irregularly distributed full-waveform LiDAR data.

Therefore, we propose a novel deep learning method to solve semantic segmentation tasks for full-waveform LiDAR data (Figure 1). Our network consists of a local feature extraction method and a hierarchical global feature extraction method because the values of the waveforms are divided into two groups to easily classify using only local features (local class) and using global features (global class). As a local feature extraction method for local class, we use a simple waveform-aware 1D CNN with a pointwise class prediction. As a global feature extraction method for global class, low- to high-level feature extraction with a waveform-aware 1D CNN is used to extract local characteristic features of full-waveform LiDAR data comprising geometric information and waveforms. To achieve low- to high-level feature extraction, we further develop a hierarchical FCN using downsampling and upsampling blocks to hierarchically assemble spatial information and waveforms with a pointwise class prediction. We simply use an ensemble learning method with these blocks to predict the final prediction of each point. Additionally, ALS data generally include an imbalanced class distribution, which we consider as the class-weighted loss function for optimization. Our network can be trained in an end-to-end manner, and we can directly predict the classification labels for all input points in one forward pass without converting full-waveform LiDAR data to images or voxels. Our main contributions are as follows:

- We introduce a novel waveform-aware convolutional method that directly applies convolutions on irregular full-waveform LiDAR data to extract waveform features.
- We develop an encoder-decoder-based network with a global module including downsampling and upsampling blocks with a skip connection and waveform-aware convolutional operations and a local module with only waveform-aware convolutional operations.
- We eliminate the requirement of expensive calculations of handcrafted features and achieve superior performance on a benchmark dataset without any conversion of full-waveform LiDAR data to 2D images or voxels.

The remainder of this paper is organized as follows. In Section 2, we review studies on 3D deep learning and automatic analysis methods for full-waveform LiDAR data. The proposed method is described in Section 3. In Section 4, we detail the conducted experiments for the verification of the semantic segmentation performance of the proposed network. We further discuss the effectiveness of the proposed method in this section. Finally, the paper is concluded in Section 5.

2 RELATED STUDY

2.1 3D Deep Learning

Deep learning is widely used in various fields, such as natural language processing [21], speech recognition [17], image processing [15], and point cloud processing [3, 12, 20, 48, 51], and others. Goodfellow et al. [11]. CNNs are among the most commonly used deep learning methods in the image-processing domain. They have achieved extremely promising results in various 2D image recognition tasks, object detection, and semantic segmentation. However, dealing with non-Euclidian, unordered, and irregular 3D point clouds is quite challenging. Deep learning-based approaches for handling 3D point clouds are divided into two methods.

The Euclidean method transforms the 3D point cloud into Euclidean data, such as 2D images or 3D voxels and is widely used for classical problems. The converted 2D images are easily fed into a 2D CNN to classify each pixel [46, 46, 47, 53]. For example, Su et al. [35] proposed to first generate multiple 2D rendered images of 3D shapes, followed by a conventional 2D CNN to extract features from each view. A view-pooling layer was further proposed to fuse information from multiple views and improve the classification performance. However, the loss of spatial information in the process of converting 3D to 2D images was notable. Therefore, some simple 3D deep learning approaches for point cloud classification using voxel-grid-based classification methods were proposed [13, 32, 43]. Similar to 2D-based methods, voxel-based methods lack point cloud data.

To address the loss of information, non-Euclidean point clouds were proposed. PointNet [29] is one of the first methods that directly handles point cloud data using a weight-shared multilayer perceptron (MLP) and an order-invariant global context from a max-pooling layer as the symmetric function. Considering the significant success of PointNet [29] and PointNet++ [30], recent 3D point cloud classification and semantic segmentation methods have been built upon PointNet architectures.

For 3D point clouds acquired via ALS, Yousefhussien et al. [49] proposed a 1D FCN-based method. This method used two input data, point clouds, and spectral features converted from 2D images, which classified each point using an end-to-end process. Wang et al. [39] created a novel pooling-layer method to classify point clouds comprised of three steps. First, they extracted pointwise features using a weight-shared MLP similar to PointNet [29]. Second, a spatial max-pooling layer was employed to extract features. Finally, another MLP layer was used to classify each layer. Wen et al. [41] proposed a multiscale FCN that considered direction. Winiwarter et al. [42] investigated the applicability of PointNet++ for semantic classification of not only benchmark data but also actual airborne LiDAR point clouds. Additionally, a task-specific study regarding the extraction of ground information from forested areas using airborne LiDAR point clouds and a dynamic graph CNN [40]-based network [50] was proposed with benchmark competition.

2.2 Full-Waveform LiDAR Data Analysis

Full-waveform LiDAR is highly advantageous to 3D point cloud semantic segmentation tasks [8, 25, 28, 31]. There are two basic methods: handcrafted feature-based and data-driven waveform-based methods.

Handcrafted feature-based methods can easily classify objects using full-waveform LiDAR data and their handcrafted features [10]. For example, a rule-based algorithm was used for classification tasks [2, 36]. Other methods are based on classic machine learning that uses handcrafted features, such as nonlinear classification and support vector machine (SVM) classifiers [24], which have been widely used with point cloud classification tasks using handcrafted features from full-waveform LiDAR data [6, 7, 14, 23, 54]. Furthermore, for land-use classification tasks, Wang et al. [37] demonstrated the importance of spatial distributional and handcrafted features of waveforms. Additionally, Yuan et al. [18] combined the ensemble method with an SVM model to improve classification capability [18]. Other researchers used the multimodal method to combine hyperspectral images and full-waveform LiDAR data [22, 38]. However, all these machine learning algorithms depended greatly upon handcrafted features from full-waveform LiDAR data used in rule-based or machine learning algorithms.

The data-driven methods overcome the need for handcrafted features. One of the first approaches without handcrafted features was proposed by Maset et al. [26]. This method used the self-organization map to solve the unsupervised classification of waveforms into three land covers (grass, trees, and roads). Recently, other novel data-driven algorithms based on CNN organized full-waveform LiDAR data into six classes (ground, vegetation, building, power line, transmission tower, and street path) [55]. The CNN-based methods included a 1D CNN and 2D FCN. The simple 1D CNN was first used to translate each waveform into a class probability vector. By leveraging the coordinates of the points associated with the waveforms, the output vector generated by the first 1D CNN and its height information was mapped to 2D image data, classifying each pixel with the FCN. They used the two-step method because individual learning for each waveform was not performed. However, the 2D FCN method [55] suggests that the spatial learning method of full-waveform LiDAR data is advantageous. As a spatial learning method for waveform, an autoencoder-based representation learning method was presented by Shinohara et al. [33]. This method directly dealt with spatially distributed full-waveform LiDAR data, which consists of geometric information and waveform, using a PointNet-based network without any conversion process to images or voxels. Their encoder extracted the compact representation as a latent vector of each input data, and their decoder reconstructed the spatial distribution and the waveform of input data with low error. However, they only demonstrated the effectiveness of the spatial learning method of full-waveform LiDAR data; they did not show a specific classification capability.

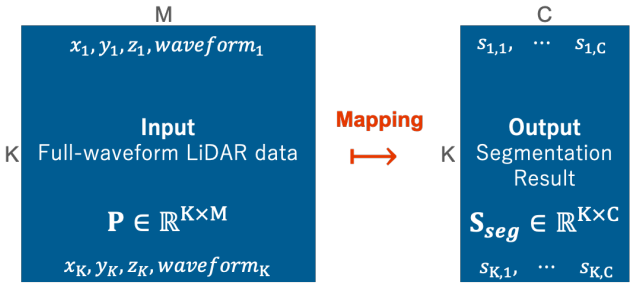


Figure 2: Problem statement and notation. Our task is mapping input full-waveform LiDAR data $P \in \mathbb{R}^{K\times M}$ with point geometry and waveform into segmentation results $S_{seg} \in \mathbb{R}^{K\times C}$

3 PROPOSED METHOD

3.1 Problem Definition and Notation

Figure 2 shows our problem definition and related notations for semantic segmentation tasks that include full-waveform LiDAR data. Given a set of input full-waveform LiDAR data, $\{p_i\}_{i=1}^{K}$ with $p_i \in \mathbb{R}^M$, we can formulate the input of the network as a matrix, $P \in \mathbb{R}^{K\times M}$, where M denotes the input feature dimension, and K denotes the total number of input points. The input data of each point consist of both geometric and radiometric waveforms, *i.e.* 3D coordinates (i.e., x, y, z), and the sequential power of the return values. In this study, we trained the network mapping into a matrix, $S_{seg} \in \mathbb{R}^{K\times C}$, from P. Here, S_{seg} represents the segmentation results, and C is the number of classes.

3.2 Feature Extraction for Waveform

We use a convolutional operation as a feature extractor for the waveform or as a feature vector from the waveform of local class. Here, we describe a convolutional operation commonly used for 3D point cloud processing. Formally, continuous convolutions are defined as

$$(f * g)(p_i) = \int_{-\infty}^{+\infty} f(p_j) \cdot g(p_i - p_j)\ dp_j, \tag{1}$$

with the continuous feature function, $f : \mathbb{R}^D \to \mathbb{R}$, converting a feature-value to every D-dimensional position, $p_j \in \mathbb{R}^d$, and the continuous kernel function, $g : \mathbb{R}^D \to \mathbb{R}$, mapping a relative position to a kernel weight. The convolutional operation is commutative

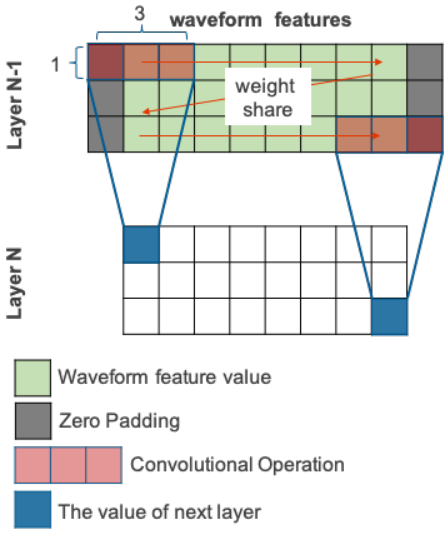


Figure 3: Simple scheme of a one-dimensional (1D) convolutional operation. The convolution outputs from the 1×3 filter are represented in blue

(i.e., $f * g = g * f$). In most real situations, the feature function, f, handles only a limited number, N, of point positions, where p_n are observed. Using Monte Carlo integration, the continuous convolution can then be approximated as

$$(f * g)(p_i) \approx \frac{1}{N} \sum_{n=1}^{N} f(p_n) \cdot g(p_i - p_n), \tag{2}$$

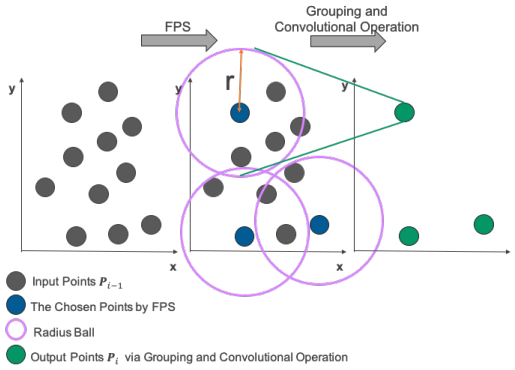
where the infinite kernel function, $g(\cdot)$, is approximated using a learned parametric function, commonly implemented as a neural network (NN):

$$g(p; \theta) = \mathrm{NN}(p; \theta), \tag{3}$$

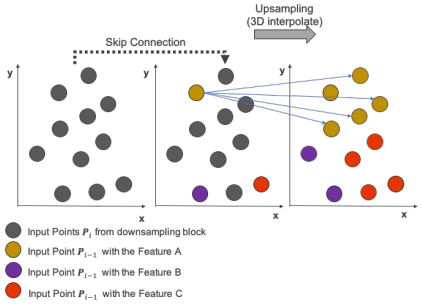
where θ is a set of learnable parameters, and NN is the neural network. This neural network is defined as a 1D CNN having a 1×3 filter widely used for audio signal processing or sequential data analysis, as shown in Figure 3.

3.3 Hierarchical Feature Extraction

Hierarchical feature extraction is an important deep learning technique that uses a CNN for image recognition tasks. Stacked local feature learning, which is used in CNN, provides a large receptive field of high-level features from the stack of local features. It is essential to design network architectures with enough receptive fields to easily discriminate classes using global context information. Pooling is a popular downsampling technique for gathering large amounts of context data for CNNs. The most popular method is farthest-point sampling (FPS), which is used in PointNet++ [30]. Leveraging the capability of PointNet++ [30] to learn high-level features from hierarchical feature extraction, we built a model with downsampling (shown as the upper side of Figure 4) and upsampling blocks (shown as the lower side of Figure 4). Downsampling blocks produce features for full-waveform LiDAR data that are sparser and more complex, whereas upsampling blocks interpolate



(a) Subsampling and grouping



(b) 3D interpolation

Figure 4: Illustration of our downsampling and upsampling block. The downsampling block produces lower resolution points with grouping and convolutional operations. The upsampling block produces the use of 3D interpolation using higher resolution points from downsampling points

the sparse set back to the original points introduced by PointNet++ [30]. We explain each block below.

Figure 4 shows our downsampling block. This block deals with input comprising full-waveform LiDAR data, $P = \{p_1 ..., p_n | \ p_i = (x_i, y_i, z_i, w_i)\}$, where $x_i, y_i, and z_i$ are 3D coordinates of the point, and w_i is the waveform in the input layer or feature vector in middle layers. The output of each downsampling block is a new set, $\hat{P} \subseteq P$, which contains a subsampled number of points (K). In other words, the number of points for the i-th downsampling block, K_i, is less than or equal to K_{i-1}. To extract the new point set, P_i, we use the FPS method. After selecting the subsampled points, we create groups for points in their respective local regions to feed them to our convolutional operation. We define local regions using a radius ball query with a hyperparameter on the search radius.

Figure 4 shows our upsampling block. This block deals with two different dimensional sets, P_i and P_{i-1}, which correspond to the output of the i-th and $(i-1)$-th downsampling blocks. This block produces a point set, $\bar{P} \supseteq P_i$), which includes the same points as those in P_{i-1}. However, it has a different feature vector. During the upsampling process, we interpolate the set of points, P_{i-1}, from known points, P_i. Additionally, we use the skip connection (shown as arrows in Figure 4) to concatenate the feature vectors in P_{i-1} with those of the interpolated set, P_{i-1}, for 1×1 convolution and extract

high-level features with a low-level-feature vanished downsampling process.

3.4 Network Architecture

Figure 5 shows our semantic segmentation network architecture for full-waveform LiDAR data. Our network consists of a local module using a simple 1D CNN and a global module using a simple 1D CNN and an encoder-decoder architecture.

The local module is shown on the upper side of Figure 5. Our local module extracts features individually from all waveforms as input and outputs a semantic segmentation result classified point by point. During local feature extraction, three convolutional operations with a 1×3 filter are applied to map into the feature space from the data space with a waveform. In this process, the number of each feature map is 256, 256, and 256. Moreover, we use an additional 1×1 convolution layer to classify each feature from three 1×3 convolutional operations. Finally, we can obtain the output matrix S_{seg}^{local} with $K \times C$.

Our encoder-decoder-based global module is shown on the lower side of Figure 5. In this figure, the encoder is shown as the blue rectangle, and the decoder is shown as the red rectangle. Our global module takes the 3D coordinates and the waveform as input and outputs a semantic segmentation result classified point by point. During encoding, three downsampling blocks are applied to reduce the size of the point set, K, to 8,192, 4,096, and 2,048. Then, during decoding, three upsampling blocks are utilized to generate dense feature prediction. The number of points is increased to 2,048, 4,096, and 8,192 in each upsampling block to reconstruct the original points. We use the convolutional operation after each downsampling operation and each upsampling operation. As a default setting, the search radius, r, for each sampling level is set to 1, 5, and 15 from bottom to top. Finally, the point features of the last upsampling block are input into a fully connected layer to produce a semantic segmentation result for all input points. Moreover, to incorporate the low-level information during the downsampling stage, we use skip connections between the same dimensional downsampling and upsampling blocks. The low-level features from the downsampling stage are concatenated with the feature matrix of the same point-set size from the upsampling stage. Other hyperparameters for our network are set as follows: feature maps for each downsampling stage are 256, 512, and 1,024 from bottom to top; feature maps for each upsampling stage are 1,024, 512, and 256 from top to bottom; and the numbers of neighbors in the convolutional operation are 16, 64, and 128 from bottom to top. Finally, we can obtain the output matrix S_{seg}^{global} with $K \times C$.

During the training process, the local module and global module are trained simultaneously.

3.5 Optimization

For 3D data processing, the distribution of each class has a problem of high imbalance. Training directly on an imbalanced dataset critically and negatively influences the overall performance of neural networks. To address this issue, we add a class-existence-aware weight coefficient for each class to the loss function. This approach is used to assist our model in learning minor classes. The balance weight for each class is determined by the logarithm function of the percentage of each class shown below.

$$\lambda_c = \frac{1}{\ln(\alpha + \frac{K_c}{\sum_{c=1}^{C} K_i})}. \tag{4}$$

Here, λ_c refers to the weight of the c-th class, P_c represents the number of points of the c-th category, C denotes the total number of classes, and α denotes the coefficient for class balance. After integrating the class-balance weights, our final loss function is defined as follows:

$$\mathcal{L}_{seg} = \sum_{i=1}^{K} \sum_{c=1}^{C} \lambda_c t_{i,c} \log s_{i,c}, \tag{5}$$

where K is the number of points, $t_{i,c}$ is the ground-truth label, $s_{i,c}$ is the predicted probability of the i-th point for the c-th class, and λ_c denotes the balance weight for class c.

The total loss is a linear summation of the segmentation results of the local module and global module, as shown below:

$$\mathcal{L}_{total} = \mathcal{L}_{seg}^{local} + \mathcal{L}_{seg}^{global}. \tag{6}$$

Here, $\mathcal{L}_{seg}^{local}$ is loss for the segmentation result from the local module, and $\mathcal{L}_{seg}^{global}$ is loss for the segmentation result from the global module.

3.6 Prediction for Test Data and Evaluation

During the test stage, the trained model directly deals with all points in each patch for semantic segmentation. Since two classification results are output from the local module (S_{seg}^{local}) and global module (S_{seg}^{global}), it is necessary to integrate these two outputs to predict final segmentation result (S_{seg}) (green rectangle in Figure 5). For the local classes, we assume that the local features are effective for classification, and we adopt the local module if the output from the local module via the softmax function is more than 0.5; otherwise, we adopt the global module. Finally, we can easily merge the class predictions from each test patch to make the segmentation results for all of test data.

Metrics for evaluating test-data precision are recall, precision, and F1 score. These metrics are widely used to evaluate the performance of semantic segmentation. Precision is the indicator for overdetection, and recall is the indicator for how many truly relevant results are returned. The F1 score considers the precision and recall of the classification model, becoming generally more suitable for cases where the categories are unevenly distributed. The precision, recall, and F1 score for each category are defined as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8}$$

$$\text{F1 score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision+recall}}, \tag{9}$$

where TP (true positive) indicates the positive data that were correctly classified by the trained model, FP (false positive) indicates negative data that were incorrectly classified as positive, and FN (false negative) indicates positive data that were misclassified as negative.

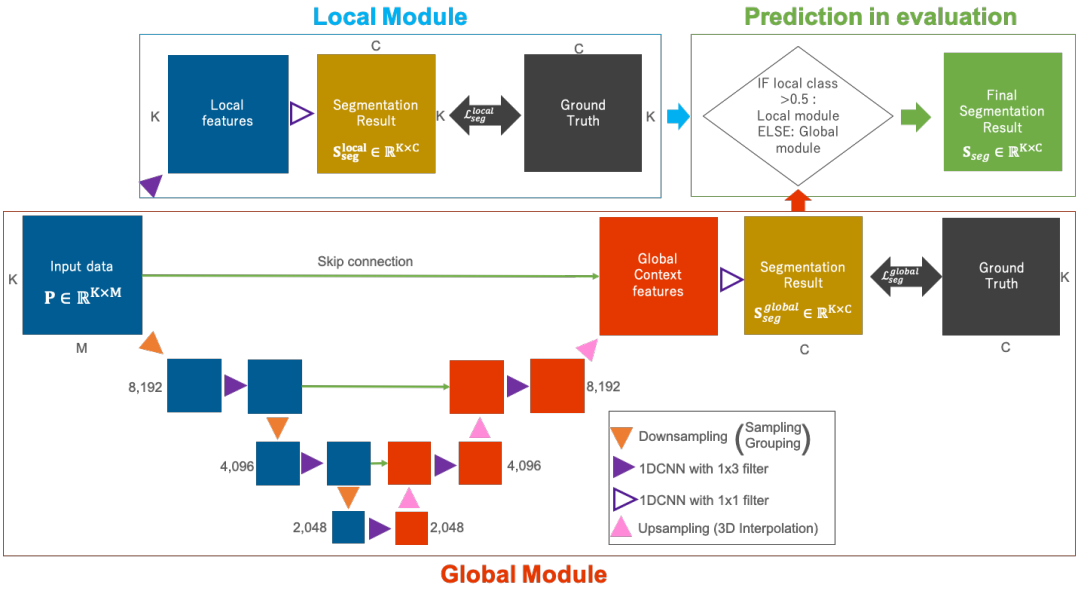


Figure 5: Architecture overview of our proposed model for solving semantic segmentation tasks for full-waveform LiDAR data. Our method consist of local module and global module. Local module is simple 1D CNN-based network. Global module begins with an encoder network used to extract high-level semantic features by downsampling (i.e., sampling and grouping) and waveform-aware convolutional operations (i.e., 1D CNN). A decoder network with an upsampling block is used to predict pointwise classification results. For evaluation, two classification results from the local module (S_{seg}^{local}) and global module (S_{seg}^{global}) are integrated to predict final segmentation result(S_{seg})

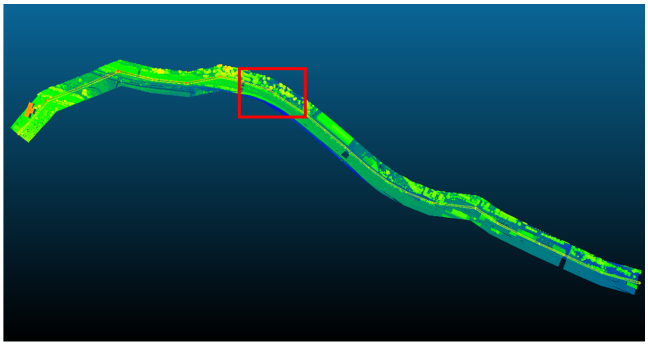


Figure 6: Example of the dataset used in this study. Points are colored by height. The red rectangle indicates the area of the test data

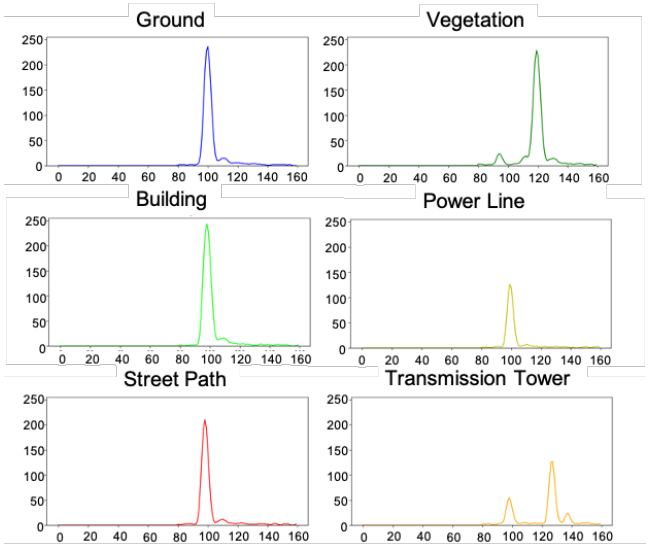


Figure 7: The mean value of the waveform of each class was calculated from 100 random samples

4 EXPERIMENTAL RESULT AND DISCUSSION

4.1 Dataset

For training, validation, and testing, we used the dataset acquired by Riegl LMS-Q780 full-waveform ALS [55]. The target area included both natural surfaces and artificial objects (Figure 6). The waveform was described by a vector of 160 values if its shortest signals were padded with zeros to reach this length. The geometric information of the point and the label shows the class to which each point belongs. These labels were assigned manually among six classes: *ground*, *vegetation*, *building*, *power line*, *transmission tower*, and *street path*. Figure 7 shows the mean value of the randomly

Table 1: Training and test data

		Train		Test	
Local	Class	num.	%	num.	%
	Ground	1,787,352	20.4	193,070	18.1
✓	Vegetation	4,719,634	53.9	765,327	71.7
	Building	1,514,486	17.3	49,138	4.6
✓	Power Line	71,978	0.8	8,151	0.8
✓	Trans. Tower	32,008	0.4	1,829	0.2
	Street Path	633,606	7.2	49,580	4.6

sampled waveform from each class. We can see that *ground*, *building*, and *street path* have similar shapes and vegetation profiles, and *transmission tower* has a characteristic shape. The dataset comprises more than nine million points divided into subsets. One subset is separated as a test dataset as shown red rectangle in Figure 6. Test data are used only during the testing phase. A key aspect of this dataset is its imbalanced distribution of each class. Table 1 shows in detail the point distribution over the classes.

We assumed that waveform data are divided into local class and global class. The local class is that local waveform shapes contribute to classification performance. The global class is that global features contribute to classification performance. We were able to see from Figure 7 that the *vegetation, power line*, and *transmission tower* classes are well characterized by the shape of the waveform. However, we were able to see from the figure that the *ground, building*, and *street path* classes are difficult to identify only in the form of waveforms. Thus, we determined *vegetation, power line*, and *transmission tower* as the local classes to prioritize the output from the local module.

The original data was a large area containing millions of order points. We cannot directly handle such large data using our model during the network training process owing to the limited graphical processing unit (GPU) memory. Thus, we split the large training area into smaller patches for training data. We divided the training scenes into fixed point patches with 50,000 points. Each patch contained a different number of points. During the training process, we used 5-fold cross-validation.

4.2 Experimental Setup

We implemented our network using TensorFlow [1]. For all cross-validation processes, we used the Adam optimizer with an initial learning rate of 0.001 and the same learning-rate schedule. The learning rate decayed 50% after every 10 epochs. The networks were trained using NVIDIA Tesla P100 GPUs in TSUBAME3.0. The batch size was set to 1 for each GPU. Batch normalization was applied to each layer. A dropout rate of 0.55 was used at the second 1D CNN layer of the last prediction block. We trained our models end-to-end from scratch using random initialized parameters.

4.3 Training Results

Our network was trained using 50,000 points per batch. Hence, the total number of patches for training was approximately 100. We employed a five-fold cross-validation dataset to train the proposed model until convergence. This training process required 6 hours per epoch.

The final semantic segmentation results of the test data are shown in Figure 8. The proposed model successfully generated the correct label predictions for most of the points from the test scenes. Qualitatively, our network tended to fail when classifying the boundary area data (as shown by the white circle). This misclassification occurred because of the lack of global information from the global module. Specifically, the reason for this is that a sufficiently wide receptive field cannot be acquired in a pointless region and that the training data have little data at the boundary. We also observed a tendency to classify street path classes as ground, as indicated by the pink circles.

To quantitatively evaluate classification performance, we calculated the precision, recall, and F1 score of each category and listed the results in Table 2. We see that our proposed model obtained F1 scores greater than 80% for six of the categories. Furthermore, our model achieved a higher recall value for all classes than the 1D CNN-based method [55] and a higher recall value for all classes without *transmission tower* than the 2D FCN-based method [55]. The results of the quantitative evaluation of our method with previous studies show that our method, which can directly deal with full-waveform LiDAR data without converting into images, performs better than previous studies. Additionally, compared with our experiment using the PointNet [29]-based model (as shown in Appendix A.1) and PointNet++ [30]-based model (as shown in Appendix A.2) trained on the same data shown in Section 4.1, our model showed higher performance for all metrics. We have shown that our method specific to waveform is more effective than the layer design proposed for the point cloud.

Generally, the proposed method improved the mean recall value approximately 1.97 times that of the 1D CNN-based method [55], approximately 1.02 times that of the 2DFCN-based method [55], approximately 1.17 times that of our reproduced experiment of PointNet [29], and approximately 1.07 times that of our reproduced experiment of PointNet++ [30].

4.4 Ablation analysis

To show the effectiveness of the proposed method, three models were created that omitted functions from the proposed method. Since our method relied on PointNet++, these models are the PointNet++ model without waveforms (Model A), the PointNet++ model without hierarchical feature extraction of waveforms (Model B), and the PointNet++ with waveforms (Model C).

The Model C is the baseline model in this ablation analysis trained the same dataset (x, y, z, and waveform) as described in Section 4.1, but the change from our proposed method was that the local module of Figure 5 was removed, and only the global module was used. In other words, Model C was imprinted using the PointNet++ [30] architecture.

4.4.1 Waveform. To demonstrate the effectiveness of the proposed waveform information, we developed a model using only geometric data (called Model A). Here, the geometric data are the dataset with the waveform removed from the dataset described in Section 4.1. To experiment with the effects of pure waveforms, we used a primitive deep learning method. Specifically, Model A was imprinted using

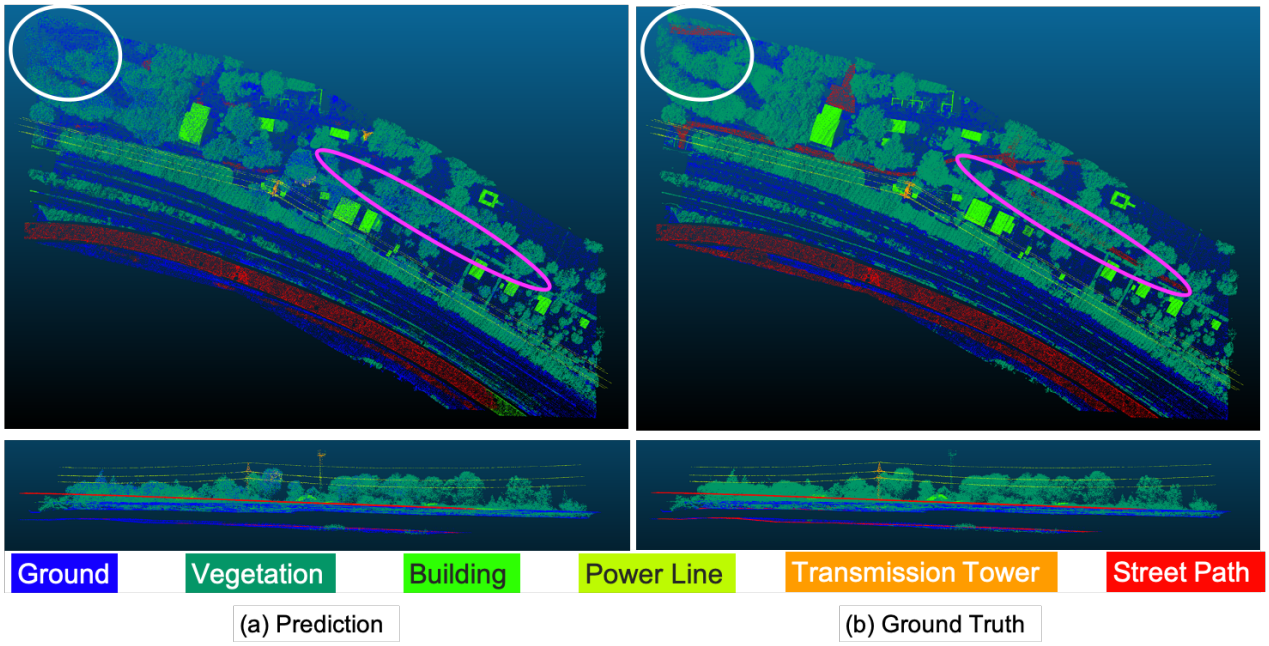


Figure 8: Qualitative results on the test dataset: (a) prediction; (b) ground truth

Table 2: Quantitative comparisons between methods. The table shows the precision, recall, and F1 score for each category

Method	Metric	Ground	Vegetation	Building	Power Line	Transmission Tower	Street Path	Mean
1D CNN [55]	Recall	0.07	0.79	0.13	0.91	0.42	0.56	0.48
2D FCN [55]	Recall	0.84	0.97	0.93	0.91	**0.88**	0.69	0.87
PointNet	Precision	0.56	0.97	0.95	0.92	0.61	0.94	0.83
	Recall	0.91	0.85	0.83	0.84	0.48	0.62	0.76
	F1 Score	0.69	0.91	0.88	0.88	0.53	0.75	0.77
PointNet++	Precision	0.56	**0.98**	**0.99**	0.97	0.52	0.96	0.83
	Recall	0.93	0.85	0.85	0.85	0.66	0.60	0.83
	F1 Score	0.70	0.91	0.91	0.91	0.58	0.74	0.80
Ours	Precision	**0.80**	**0.98**	**0.99**	**0.99**	**0.99**	**0.97**	**0.95**
	Recall	**0.94**	**0.96**	**0.95**	**0.99**	0.79	**0.71**	**0.89**
	F1 Score	**0.86**	**0.97**	**0.97**	**0.99**	**0.88**	**0.82**	**0.92**

the vanilla PointNet++ [30] architecture. We list the classification results of this model in Table 3. Comparing Model A and Model C, Model C using the waveform improved the precision value by 16%, recall value by 19%, and F1-score by 19%.

4.4.2 Hierarchical Feature Extraction. To demonstrate the effectiveness of the PointNet++-based hierarchical model, we developed another model without hierarchical feature extraction (called Model B). This Model B was imprinted using the vanilla PointNet [29] architecture. Model B, as well as our proposed method, trained the same dataset as described in Section 4.1. We list the classification results of this model in Table 3. Comparing Model B and Model C, Model C using hierarchical feature extrication improved the precision value by 16%, recall value by 24%, and F1-score by 25%.

4.4.3 Local Feature Extraction. To demonstrate the effectiveness of the local module, which can extract the local features of the waveform, we compared our proposed model (Ours) with Model C. We list the classification results of this model in Table 3. Comparing Model C and Ours, the proposed method using the local module improved the precision value by 19%, the recall value by 7%, and the F1-score by 15%.

4.5 The Local Module Effeteness for Final Result

Our method integrates the two outputs of the local module and the global module to obtain the final output for the test data. For the final output, we examined the contribution of the local module to

Table 3: Ablation analysis of our model. The boldface text indicates the model with the best performance

Model	Geometry	Waveform	Hierarchical	Local	Mean precision	Mean recall	Mean F1-score
Model A	✓	×	✓	×	0.69	0.70	0.67
Model B	✓	✓	×	×	0.69	0.67	0.64
Model C	✓	✓	✓	×	0.80	0.83	0.80
Ours	✓	✓	✓	✓	**0.95**	**0.89**	**0.92**

the three local class, *vegetation, power line*, and *transmission tower*, in which we assumed that the local feature is effective for classification. To investigate the contribution of the local module, we calculated the acceptance ratio of the output obtained from the local module to the final output4. As shown in Table 4, the local module showed a high acceptance ratio in the local classes *vegetation, power line*, and *transmission tower*, as we assumed. However, the method of integrating the local module and the global module used in this paper was adopted from the local module in an aggressive manner and, therefore, was also adopted for *ground, building*, and *street path*, which we did not assume. These results suggest the need for a data-driven method to improve the rule-based integration method for learnable integration.

Table 4: Quantitative assessment of the effect of our local module. The acceptance ratio from the local module for each class for the final prediction for the test data is shown

Class	Acceptance ratio from local module
Ground	0.15
Vegetation	0.82
Building	0.09
Power line	0.89
Transmission tower	0.96
Street path	0.13

5 CONCLUSION

In this paper, we proposed a novel deep learning model for the task of full-waveform LiDAR data, including point clouds and waveforms, semantic segmentation without converting point clouds or waveforms into images or voxels. Our model consists of a local module and global module with waveform-aware convolutional operations. Local modules are based on the proposed convolutional operation to extract local features. The global module is based on an encoder-decoder-based model using hierarchical downsampling and upsampling blocks with convolutional operations to extract global features. Experimental results on the benchmark dataset have shown F1 scores greater than 80% for six of the classes, demonstrated higher recall than previous methods for all class excepted *vegetation* class, and higher F1-scores than other networks for 3D data analysis (PointNet [29] and PointNet++[30]) for all classes. Additionally, by conducting an ablation study, the effectiveness of the three devices (waveform, hierarchical learning, and local module) incorporated in this study was demonstrated. Moreover, we showed that the acceptance ratio from the local module for the final output is high for the three classes (*vegetation, power line*, and *transmission tower*), where we assumed that the local feature is effective. We thus conclude that our model can produce a semantic classification result with a high performance by only taking the full-waveform LiDAR data as input without any conversion process.

ACKNOWLEDGMENTS

We would like to gratefully acknowledge the benchmark data owners for providing airborne full-waveform LiDAR data. This work was partially supported by KAKENHI (19H02408). The numerical calculations were performed using the TSUBAME3.0 supercomputer at the Tokyo Institute of Technology.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. http://tensorflow.org/ Software available from tensorflow.org.

[2] Cici Alexander, Kevin Tansey, Jörg Kaduk, David Holland, and Nicholas J. Tate. 2010. Backscatter coefficient as an attribute for the classification of full-waveform airborne laser scanning data in urban areas. *ISPRS J. Photogramm. Remote Sens.* 65, 5 (Sept. 2010), 423–432. DOI: https://doi.org/10.1016/j.isprsjprs.2010.05.002.

[3] Saifullahi Aminu Bello, Shangshu Yu, and Cheng Wang. 2020. Deep learning on 3D point clouds. *arXiv* (2020), arXiv-2001.

[4] Hans-Erik Andersen, Robert J McGaughey, and Stephen E Reutebuch. 2005. Estimating forest canopy fuel parameters using LiDAR data. *Remote Sens. Environ.* 94, 4 (Feb. 2005), 441–449. DOI: https://doi.org/10.1016/j.rse.2004.10.013.

[5] Peter Axelsson. 2000. DEM generation from laser scanner data using adaptive TIN models. *Int. Arch. Photogramm. Remote Sens.* 33, 4 (2000), 110–117.

[6] Mohsen Azadbakht, Clive Fraser, and Kourosh Khoshelham. 2015. The role of full-waveform LiDAR features in improving urban scene classification.

[7] Mohsen Azadbakht, Clive S. Fraser, and Kourosh Khoshelham. 2018. Synergy of sampling techniques and ensemble classifiers for classification of urban environments using full-waveform LiDAR data. *Int. J. Appl. Earth Obs. Geoinforma.* 73 (Dec. 2018), 277–291. DOI: https://doi.org/10.1016/j.jag.2018.06.009.

[8] Vesna Ducic, Markus Hollaus, Andreas Ullrich, Wolfgang Wagner, and Thomas Melzer. 2006. *3D vegetation mapping and classification using full-waveform laser scanning.* na.

[9] Liviu Theodor Ene, Erik Næsset, Terje Gobakken, Ole Martin Bollandsås, Ernest William Mauya, and Eliakimu Zahabu. 2017. Large-scale estimation of change in aboveground biomass in miombo woodlands using airborne laser scanning and national forest inventory data. *Remote Sens. Environ.* 188 (Jan. 2017), 106–117. DOI: https://doi.org/10.1016/j.rse.2016.10.046.

[10] Karolina D. Fieber, Ian J. Davenport, James M. Ferryman, Robert J. Gurney, Jeffrey P. Walker, and Jorg M. Hacker. 2013. Analysis of full-waveform LiDAR data for classification of an orange orchard scene. *ISPRS J. Photogramm. Remote Sens.* 82 (Aug. 2013), 63–82. DOI: https://doi.org/10.1016/j.isprsjprs.2013.05.002.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT Press. http://www.deeplearningbook.org.

[12] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. 2019. Deep learning for 3D point clouds: A survey. *arXiv preprint arXiv:1912.12033* (2019).

[13] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D. Wegner, Konrad Schindler, and Marc Pollefeys. 2017. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. IV-1-W1. 91–98.

[14] Bernhard Höfle, Markus Hollaus, and Julian Hagenauer. 2012. Urban vegetation detection using radiometrically calibrated small-footprint full-waveform airborne LiDAR data. *ISPRS J. Photogramm. Remote Sens.* 67 (Jan. 2012), 134–147. DOI: https://doi.org/10.1016/j.isprsjprs.2011.12.003.

[15] Licheng Jiao and Jin Zhao. 2019. A survey on the new generation of deep learning in image processing. *IEEE Access* 7 (Nov. 2019), 172231–172263. DOI: https://doi.org/10.1109/ACCESS.2019.2956508.

[16] Martin Kada and Laurence McKinley. 2009. 3D building reconstruction from LiDAR based on a cell decomposition approach. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 38, Part 3 (Sept. 2009), W4.

[17] Akshi Kumar, Sukriti Verma, and Himanshu Mangla. 2018. A survey of deep learning techniques in speech recognition. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*. 179–185. DOI: https://doi.org/10.1109/ICACCCN.2018.8748399.

[18] Xudong Lai, Yifei Yuan, Yongxu Li, and Mingwei Wang. 2019. Full-waveform LiDAR point clouds classification based on wavelet support vector machine and ensemble learning. *Sensors* 19 (07 Jul. 2019), 3191. DOI: https://doi.org/10.3390/s19143191.

[19] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. 2018. PointCNN: Convolution on X-transformed points. In *Advances in Neural Information Processing Systems*. 820–830.

[20] Weiping Liu, Jia Sun, Wanyi Li, Ting Hu, and Peng Wang. 2019. Deep learning on point clouds and its application: A survey. *Sensors* 19, 19 (Oct. 2019), 4188. DOI: https://doi.org/10.3390/s19194188.

[21] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. 2017. A survey of deep neural network architectures and their applications. *Neurocomputing* 234 (Apr. 2017), 11–26. DOI: https://doi.org/10.1016/j.neucom.2016.12.038.

[22] Shezhou Luo, Cheng Wang, Xi Xiaohuan, Hongcheng Zeng, Dong Li, Shaobo Xia, and Pinghua Wang. 2015. Fusion of airborne discrete-return LiDAR and hyperspectral data for land cover classification. *Remote Sens.* 8 (12 Dec. 2015), 3. DOI: https://doi.org/10.3390/rs8010003.

[23] Lian Ma, Mei Zhou, and Chuanrong Li. 2017. LAND covers classification based on random forest method using features from full-waveform LiDAR data. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* XLII-2/W7 (Sep. 2017), 263–268. DOI: https://doi.org/10.5194/isprs-archives-XLII-2-W7-263-2017.

[24] Clément Mallet, Frédéric Bretar, Michel Roux, Uwe Soergel, and Christian Heipke. 2011. Relevance assessment of full-waveform LiDAR data for urban area classification. *ISPRS J. Photogramm. Remote Sens.* 66, 6, Supplement (Dec. 2011), S71–S84. DOI: https://doi.org/10.1016/j.isprsjprs.2011.09.008. Advances in LiDAR Data Processing and Applications.

[25] Clément Mallet, Uwe Soergel, and Frédéric Bretar. 2008. Analysis of full-waveform LiDAR data for classification of urban areas.

[26] Eleonora Maset, Roberto Carniel, and Fabio Crosilla. 2015. Unsupervised classification of raw full-waveform airborne LiDAR data by self organizing maps. In *Image Analysis and Processing—ICIAP 2015*, Vittorio Murino and Enrico Puppo (Eds.). Springer International Publishing, Cham, 62–72.

[27] Domen Mongus and Borut Žalik. 2013. Computationally efficient method for the generation of a digital terrain model from airborne LiDAR data using connected operators. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 7, 1 (May 2013), 340–351. DOI: https://doi.org/10.1109/JSTARS.2013.2262996.

[28] Amy L Neuenschwander, Lori A Magruder, and Marcus Tyler. 2009. Landcover classification of small-footprint, full-waveform LiDAR data. *J. Appl. Remote Sens.* 3, 1 (Aug. 2009), 033544. DOI: https://doi.org/10.1117/1.3229944.

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017. Pointnet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 652–660.

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*. 5099–5108.

[31] Josef Reitberger, Peter Krzystek, and Uwe Stilla. 2009. Benefit of airborne full waveform LiDAR for 3D segmentation and classification of single trees. In *ASPRS 2009 Annual Conference*. 1–9.

[32] S. Schmohl and U. Sörgel. 2019. Submanifold sparse convolutional networks for semantic segmentation of large-scale ALS point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inform. Sci.* IV-2/W5 (May 2019), 77–84. DOI: https://doi.org/10.5194/isprs-annals-IV-2-W5-77-2019.

[33] Takayuki Shinohara, Haoyi Xiu, and Masashi Matsuoka. 2019. FWNetAE: Spatial representation learning for full waveform data using deep learning. In *2019 IEEE International Symposium on Multimedia (ISM)*. 259–2597. DOI: https://doi.org/10.1109/ISM46123.2019.00060.

[34] Svein Solberg, Andreas Brunner, Kjersti Holt Hanssen, Holger Lange, Erik Næsset, Miina Rautiainen, and Pauline Stenberg. 2009. Mapping LAI in a Norway spruce forest using airborne laser scanning. *Remote Sens. Environ.* 113, 11 (Nov. 2009), 2317–2327. DOI: https://doi.org/10.1016/j.rse.2009.06.010.

[35] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. 2015. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE international conference on computer vision*. 945–953.

[36] Wolfgang Wagner, Markus Hollaus, Christian Briese, and Vesna Ducic. 2008. 3D vegetation mapping using small-footprint full-waveform airborne laser scanners. *Int. J. Remote Sens.* 29, 5 (Feb. 2008), 1433–1452. DOI: https://doi.org/10.1080/01431160701736398.

[37] Chisheng Wang, Qiqi Shu, Xinyu Wang, Bo Guo, Peng Liu, and Qingquan Li. 2019. A random forest classifier based on pixel comparison features for urban LiDAR data. *ISPRS J. Photogramm. Remote Sens.* 148 (Feb. 2019), 75–86. DOI: https://doi.org/10.1016/j.isprsjprs.2018.12.009.

[38] Hongzhou Wang and Craig Glennie. 2015. Fusion of waveform LiDAR data and hyperspectral imagery for land cover classification. *ISPRS J. Photogramm. Remote Sens.* 108 (Oct. 2015), 1–11. DOI: https://doi.org/10.1016/j.isprsjprs.2015.05.012.

[39] Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. 2018. Deep parametric continuous convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2589–2597.

[40] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph. (TOG)* 38, 5 (Jun 2019), 1–12.

[41] Congcong Wen, Lina Yang, Ling Peng, Xiang Li, and Tianhe Chi. 2019. Directionally constrained fully convolutional neural network for airborne LiDAR point cloud classification. *arXiv preprint arXiv:1908.06673* (2019).

[42] Lukas Winiwarter, Gottfried Mandlburger, Stefan Schmohl, and Norbert Pfeifer. 2019. Classification of ALS point clouds using end-to-end deep learning. *PFG – J. Photogramm. Remote Sens. Geoinform. Sci.* 87, 3 (01 Sep. 2019), 75–90. DOI: https://doi.org/10.1007/s41064-019-00073-0.

[43] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1912–1920.

[44] Haoyi Xiu, Takayuki Shinohara, and Masashi Matsuoka. 2019. Dynamic-scale graph convolutional network for semantic segmentation of 3D point cloud. In *2019 IEEE International Symposium on Multimedia (ISM)*. 271–2717. DOI: https://doi.org/10.1109/ISM46123.2019.00062.

[45] Bisheng Yang, Ronggang Huang, Jianping Li, Mao Tian, Wenxia Dai, and Ruofei Zhong. 2017. Automated reconstruction of building LoDS from airborne LiDAR point clouds using an improved morphological scale space. *Remote Sens.* 9, 1 (2017), 14. DOI: https://doi.org/10.3390/rs9010014.

[46] Zhishuang Yang, Wanshou Jiang, Bo Xu, Quansheng Zhu, San Jiang, and Wei Huang. 2017. A convolutional neural network-based 3D semantic labeling method for ALS point clouds. *Remote Sens.* 9, 9 (Sept. 2017), 936. DOI: https://doi.org/10.3390/rs9090936.

[47] Zhishuang Yang, Bo Tan, Huikun Pei, and Wanshou Jiang. 2018. Segmentation and multi-scale convolutional neural network-based classification of airborne laser scanner data. *Sensors* 18, 10 (Oct. 2018), 3347. DOI: https://doi.org/10.3390/s18103347.

[48] Xuanxia Yao, Jia Guo, Juan Hu, and Qixuan Cao. 2019. Using deep learning in semantic classification for point cloud data. *IEEE Access* 7 (Mar. 2019), 37121–37130. DOI: https://doi.org/10.1109/ACCESS.2019.2905546.

[49] Mohammed Yousefhussien, David J Kelbe, Emmett J Ientilucci, and Carl Salvaggio. 2018. A multi-scale fully convolutional network for semantic labeling of 3D point clouds. *ISPRS J. Photogramm. Remote Sens.* 143 (Sept. 2018), 191–204. DOI: https://doi.org/10.1016/j.isprsjprs.2018.03.018.

[50] Jinming Zhang, Xiangyun Hu, Hengming Dai, and ShenRun Qu. 2020. DEM extraction from ALS point clouds in forest areas via graph convolution network. *Remote Sens.* 12, 1 (Jan. 2020). DOI: https://doi.org/10.3390/rs12010178.

[51] Jiaying Zhang, Xiaoli Zhao, Zheng Chen, and Zhejun Lu. 2019. A review of deep learning-based semantic segmentation for point cloud (November 2019). *IEEE Access* (Dec. 2019). DOI: https://doi.org/10.1109/ACCESS.2019.2958671.

[52] Kaiguang Zhao and Sorin Popescu. 2009. LiDAR-based mapping of leaf area index and its use for validating GLOBCARBON satellite LAI product in a temperate forest of the southern USA. *Remote Sens. Environ.* 113, 8 (Aug. 2009), 1628–1645. DOI: https://doi.org/10.1016/j.rse.2009.03.006.

[53] Ruibin Zhao, Mingyong Pang, and Jidong Wang. 2018. Classifying airborne LiDAR point clouds via deep features learned by a multi-scale convolutional neural network. *Int. J. Geograph. Inform. Sci.* 32, 5 (Feb. 2018), 960–979. DOI: https://doi.org/10.1080/13658816.2018.1431840.

[54] Mei Zhou, Chunxing Li, Lingling Ma, and Hongcan Guan. 2016. Land cover classification from full-waveform LiDAR data based on support vector machines. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inform. Sci.* XLI-B3 (06 Jun. 2016), 447–452. DOI: https://doi.org/10.5194/isprs-archives-XLI-B3-447-2016.

[55] Stefano Zorzi, Eleonora Maset, Andrea Fusiello, and Fabio Crosilla. 2019. Full-waveform airborne LiDAR data classification using convolutional neural networks. *IEEE Trans. Geosci. Remote Sens.* PP (Jun. 2019), 1–7. DOI: https://doi.org/10.1109/TGRS.2019.2919472.

A BASELINE METHOD

Our method was compared within the experiment in Section 4.3 using the PointNet[29]-based model and PointNet++[30]-based model. We describe each of these methods below.

A.1 PointNet

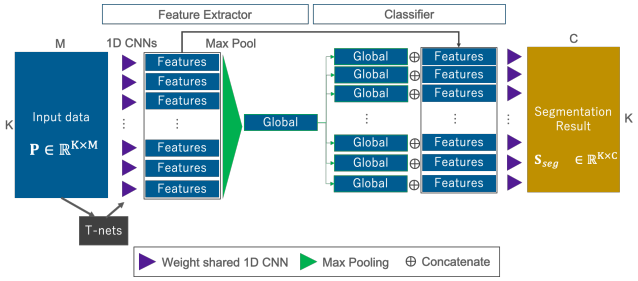


Figure 9: Architecture overview of PointNet-based model.

Figure 9 shows PointNet[29]-based model for full-waveform LiDAR data segmentation. The PointNet-based network consists of local feature extraction (purple triangle), T-Nets (grey rectangle), MaxPooling (green triangle) and classifier (right side of the figure). The local feature extraction is based on weight shared 1D CNNs. The convolutional layer extracts point wise local features from input data. T-Nets is based on learnable translation matrix. This networks provide rotation invariant feature for input data and local features. The MaxPooling layer provides order invariant or global feature from local features. Finally, classification layer calculate the class probability of each points from both of local feature and global features.

A.2 PointNet++

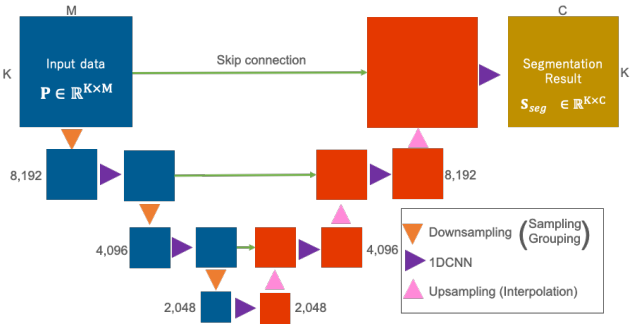


Figure 10: Architecture overview of PointNet++-based model.

Figure 10 shows PointNet++[30]-based network architecture for full-waveform LiDAR data segmentaion. The PointNet++-based network is based on an encoder-decoder architecture. The encoder-decoder architecture consist of an encoder network used to extract high-level semantic features by downsampling (i.e., sampling and grouping) and waveform-aware convolutional operations (i.e., 1D CNN) and a decoder network with an upsampling block is used to predict pointwise classification results. In other words, PointNet++-based model is our proposed method without the local module.